# Object detection of BDD100k dataset

## Exploratory data analysis:

The dataset distribution of the BDD100K shows that it is an imbalanced dataset.

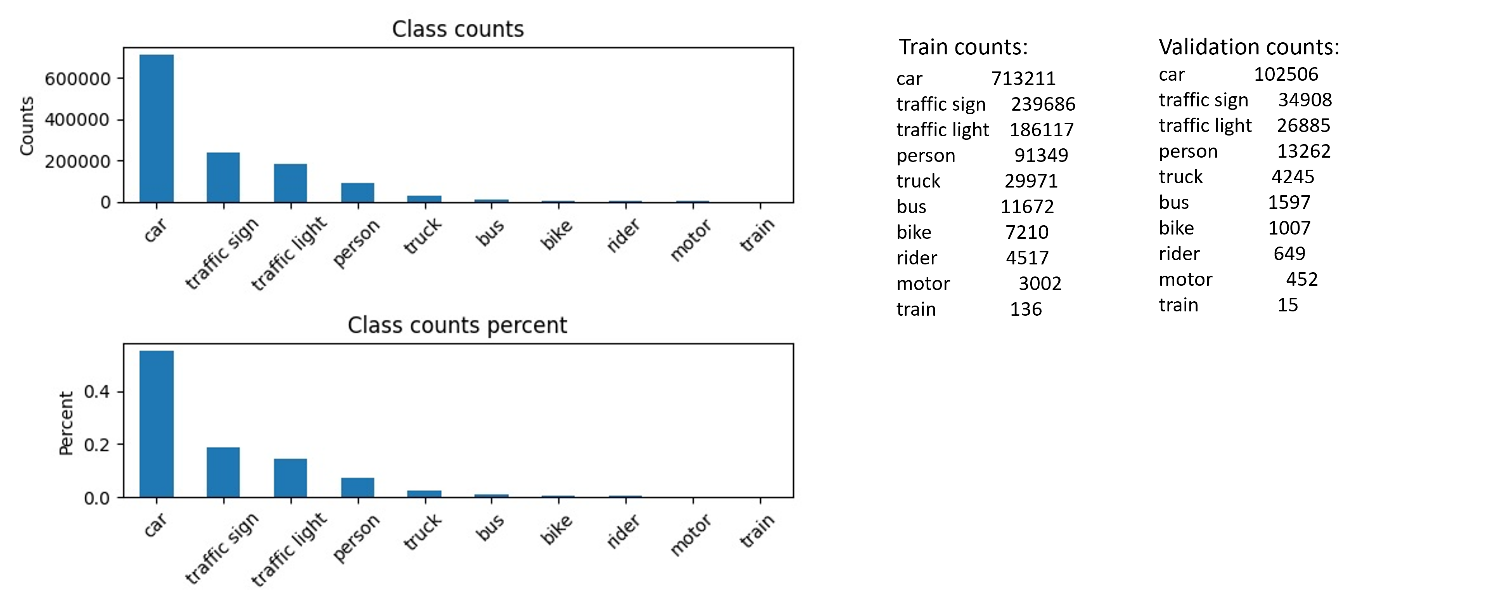


Figure Total counts of data from each class. It can be seen that there is disproportionate distribution among class examples

The average sizes i.e. area and aspect ratio () of classes also varies as given below in Figure 2:

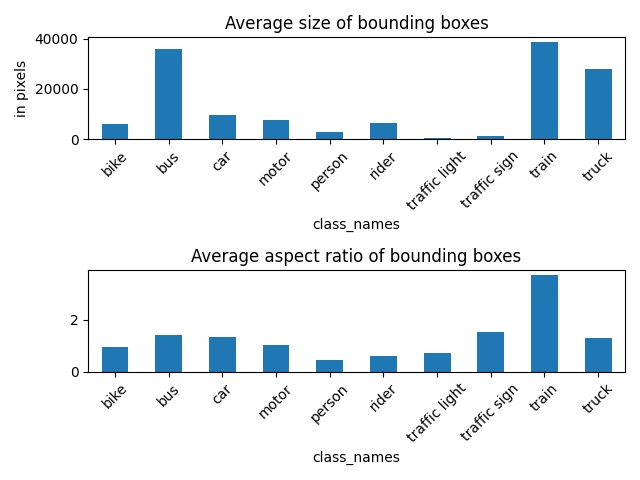


Figure Data size distribution

Since, we are using yolov5 model, which automatically assigns 9 different anchor boxes, it is not an immediate concern to tune for the size differences. The class imbalance is the major concern.

## Training:

Yolov5 model from ultralytics is used for object detection. The model is relatively light-weight and very fast for real time application. That is the reason for choice of the model. Advanced versions of yolo needs to be tested, however, currently the idea is to start simple and build iteratively. The training is done initially from the base model by freezing the first 10 layers (model backbone) and therefore transfer learning is used. The data size for some of the classes is relatively small, therefore leverage of pretrained weights is used and only later layers are finetuned. Training is done for 50 epochs with an NVIDIA GeForce 4090 GPU. Total training time is approximately 6-7 hours for the whole dataset.

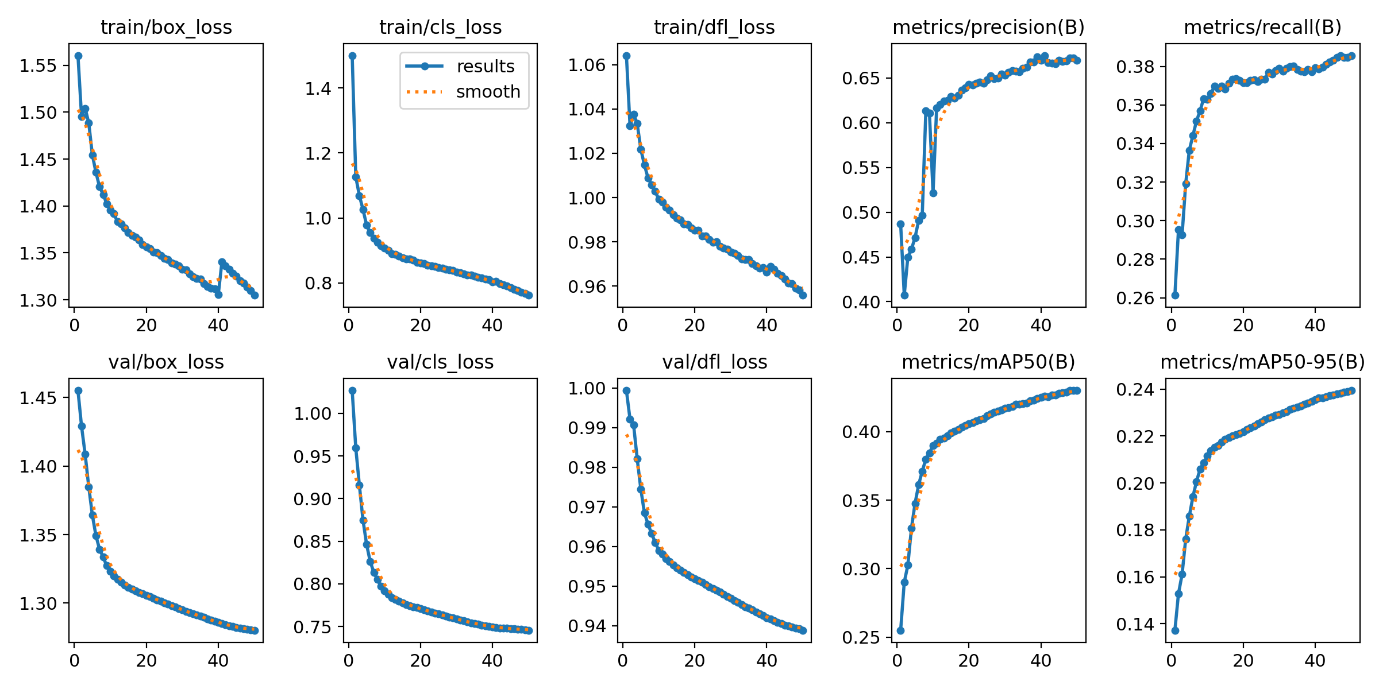
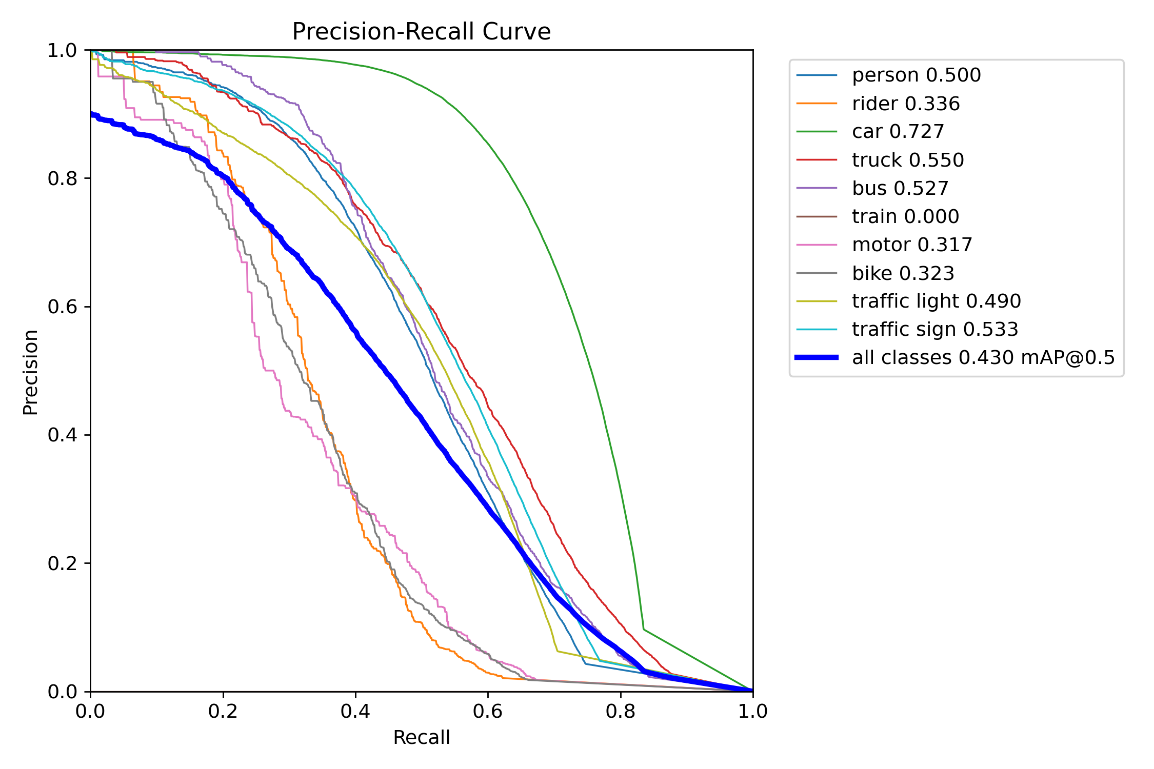


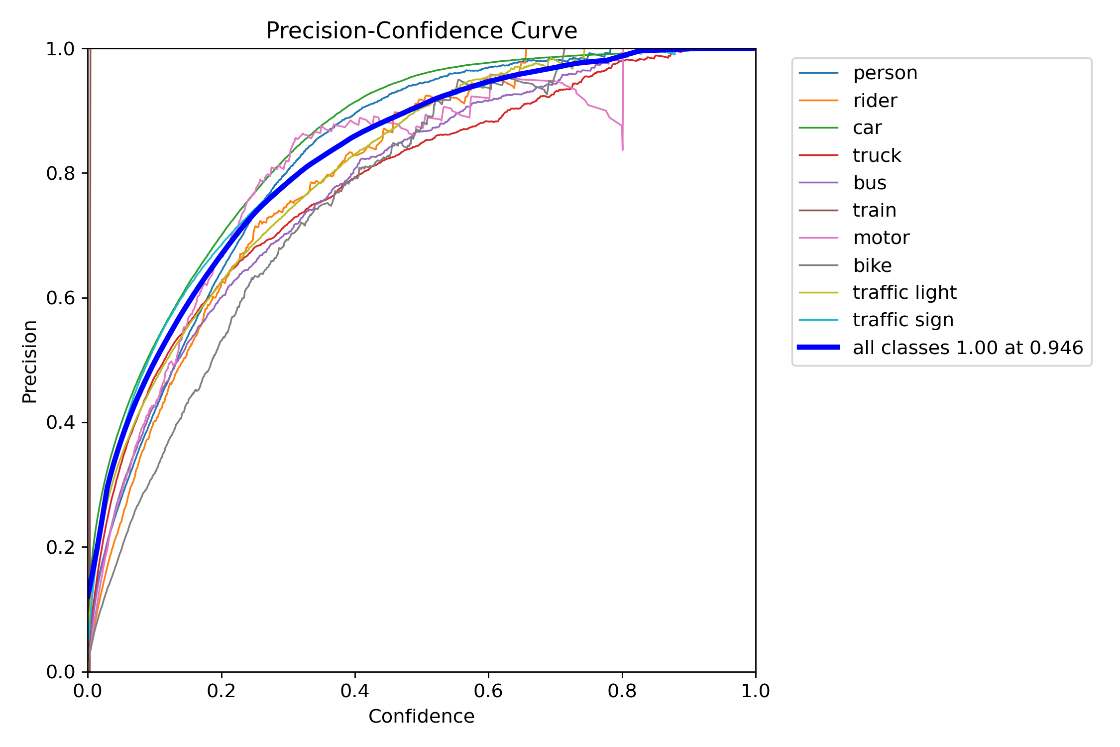
Figure Train and validation losses and essential metrics as a function of training epochs

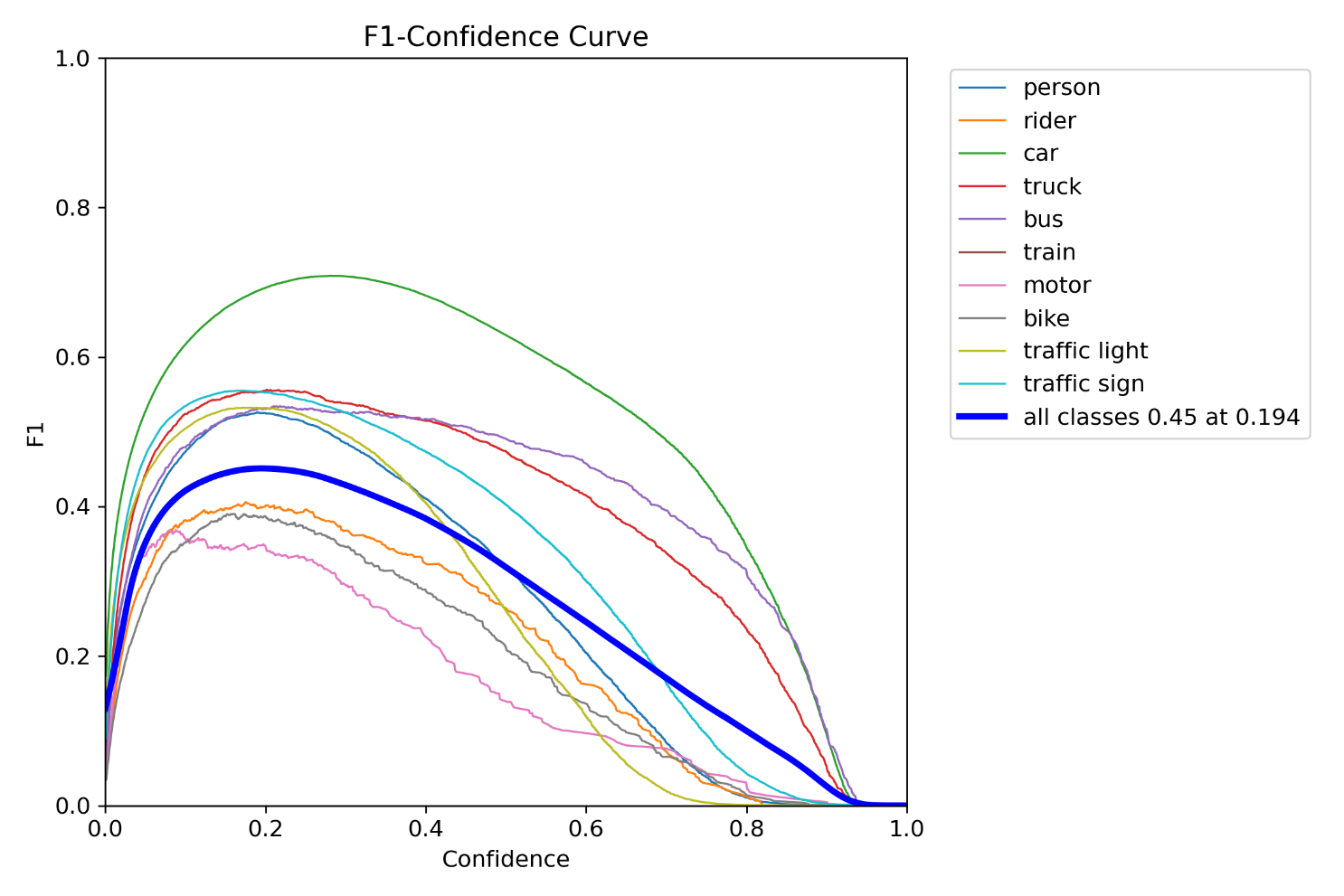
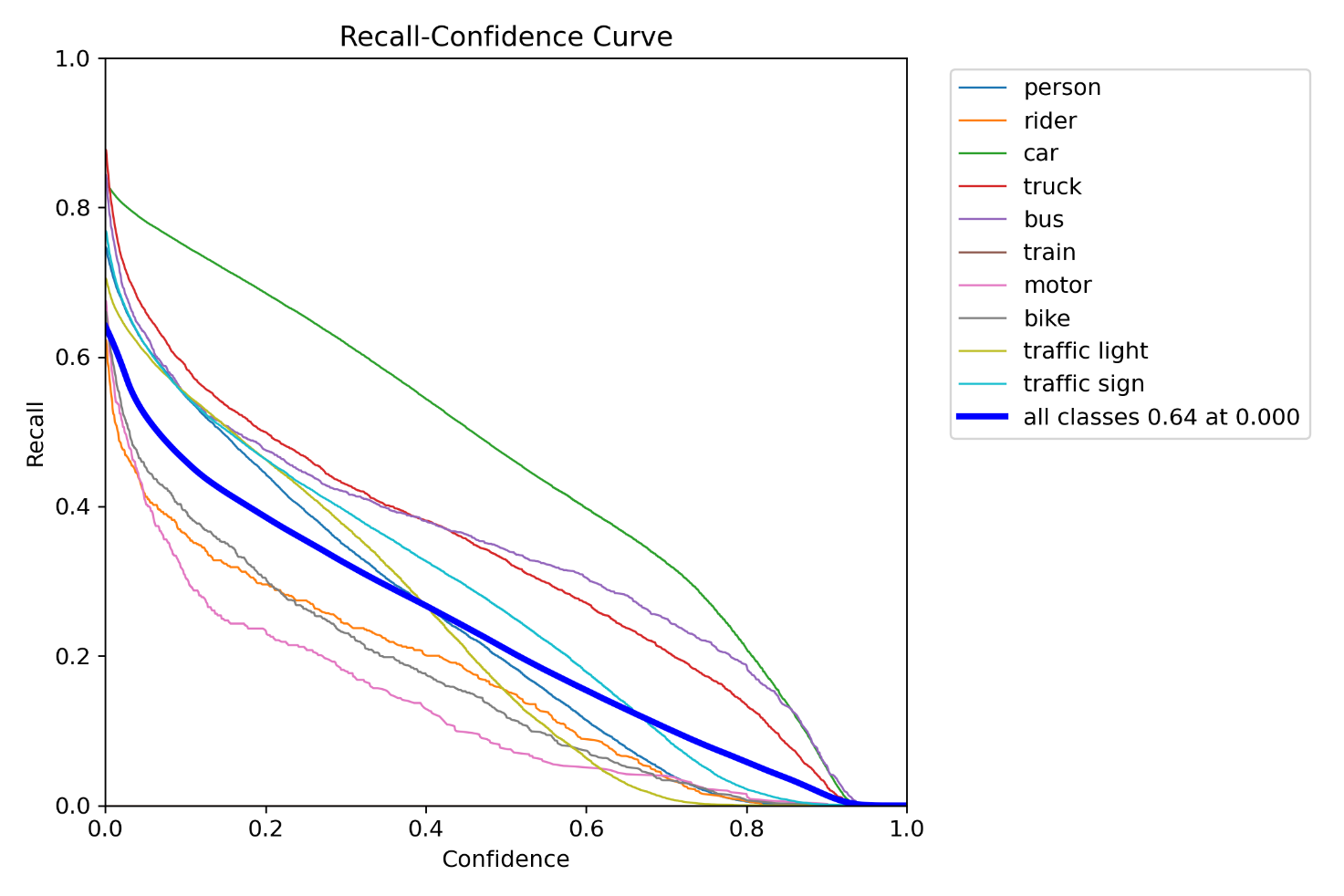
The train and validation loss decreases for every iteration and we can see that there is no saturation yet. So perhaps, the training can continue for a few more iterations.

## Evaluation:

The model is evaluated against a validation set consisting of 10,000 images. The PR (precision-recall) curve shows that the slope is inversely proportional to the data size of the classes. We can also observe higher precision for car in comparison to other classes. Classes like bike, motor, rider and train suffer from fewer examples. A similar trend can be seen for precision-confidence curve as a function of class label. The precision value generally increases with confidence as the number of false positives are decreased accordingly. The recall-confidence curve shows interesting insights. The recall decreases normally as confidence threshold increases due to decrease in true-positives and subsequent increase in false-negatives. However, the slope of this curve shows how sensitive the model is to confidence threshold and hence rejecting the detections. A steep-slope could be seen for classes other than car, again indicating the model being affected by class imbalance. The F1 score indicates a good balance between the precision and recall and we can see that a confidence threshold in the range or 0.3-0.5 provides a good balance for detection, although it is imbalanced for few classes.







Now that we have identified the effects of class imbalance on the final prediction, next steps would be to address the class imbalance.

Some of the potential steps in the order of preference:

1. Acquire more data samples for minority classes.
2. Augment the data for minority samples, use SMOTE (Synthetic Minority Over-sampling Technique) to create synthetic data by interpolation.
3. Use Focal loss and tune alpha and gamma hyperparameter to provide higher weights on hard-to-detect classes.
4. Directly modify the classification part of the loss function to account for class imbalance. Add a weighing factor as a function of class data distribution.
5. Under-sampling the majority classes leading to more even distribution, but this approach is to be considered with care as it can lead to loss of information.